Swarm Crowdsale 0.4 Source Code Review

Prepared for Swarm Fund • October 2017

V1.0

1. Table of Contents

2. Executive Summary

In September 2017, Swarm Fund Project engaged Coinspect to perform a security audit of the crowdsale, token and multi-sig wallet smart contracts, together with other inherited library contracts: OpenZeppelin and MiniMeToken. On October 4th, 2017 Sward Fund engaged Coinspect again to perform an audit of the changes in the release 0.4 of the same contracts. The contracts 0.4 were retrieved from the Swarm github repository on October 7th, 2017 from https://github.com/tokensoft/swarm-contracts/releases/tag/v0.4. The objective of the audit was to evaluate the security of the crowdsale, ERC20-compatible token and multi-sig wallet implementations with respect to the previous audited version. During the assessment, Coinspect identified one issue, of minor risk.

3. Introduction

The contract **SwarmCrowdsale.sol** is a traditional crowdsale contract that changes the token price rate at providing steps, and being pausable, and therefore owned. It dynamically mints tokens in the **SwarmToken.sol** contract, which is a ERC20-compliant contract with vesting, minting and cloning functionality, using MiniMe.sol in its core. All tokens, either pre-allocated of bought during the token sale are vested. The token cloning functionality is provided by MiniMe. MiniMe is controlled by the **SwarmCrowdsale.sol** contract during the sale, and by a

multi-signature wallet (**MultiSigWallet.sol**) after it is finalized. This wallet also receives all tokens that have not been sold during the crowdsale. Minime enables the controller to to pause and resume the trading of the token, claim other tokens or ether sent by mistake to the token contract, and generate more tokens at will.
All contracts use SafeMath methods to prevent overflows.

The release 0.4 of the contracts had the following enhancements:

- Crowdsale contract uses time stamp instead of block number
- Crowdsale can set the number of tokens already sold in the constructor and setBaseTokensSold()
- Crowdsale can pre-allocate multiple token addresses in a single call via multiPresaleMint()
- Token vesting parameters can be updated at later point via setVestingParams()

A whitebox security audit was conducted on this contract. The present report was completed on October 8th by Coinspect and includes results from the audit.

4. Findings

4.1. setBaseTokensSold can be called after the sale has started

Minor Risk

The number of tokens sold alters the sale rate. A manual modification of the number of tokens sold before and after a buy (buyTokens) using internal calls can give the buyer an unfair advantage and go unnoticed. However, this requires the collusion with the contract owner, who is the only party that can modify the amount of tokens sold.

Recommendation

Revert calls to setBaseTokensSold to when initialized != false.

5. Disclaimer

The present security audit is limited to smart contract code. It does not cover the technologies and designs related to these smart contracts, nor the frameworks and wallets that communicate

with the contracts, nor the general operational security of the company behind this project. This document should not be read as investment advice or an offering of tokens.